Microsoft

.net developer training

Create XML Web Services and Applications with Visual Studio .NET

.NET Security Advanced Topics

This module is beyond the scope of the normal 2-day seminar. It is included in the interest of completeness

Advanced Topics Agenda

- Introduction
- Evidence-based Security
- Code-based Security
- Miscellaneous security topics:
  - Cryptography Classes
  - Obfuscation
  - Type Safety
  - Security Tools

Introduction

What .NET Code Access Security is about

- Protecting the system from malicious code running in the Common Language Runtime
- Moving from monolithic trust based on user login to Role-based Trust dependant upon the code itself
  - The problems with operating code based on user login
  - The advantages of operating code with rights based on the code itself

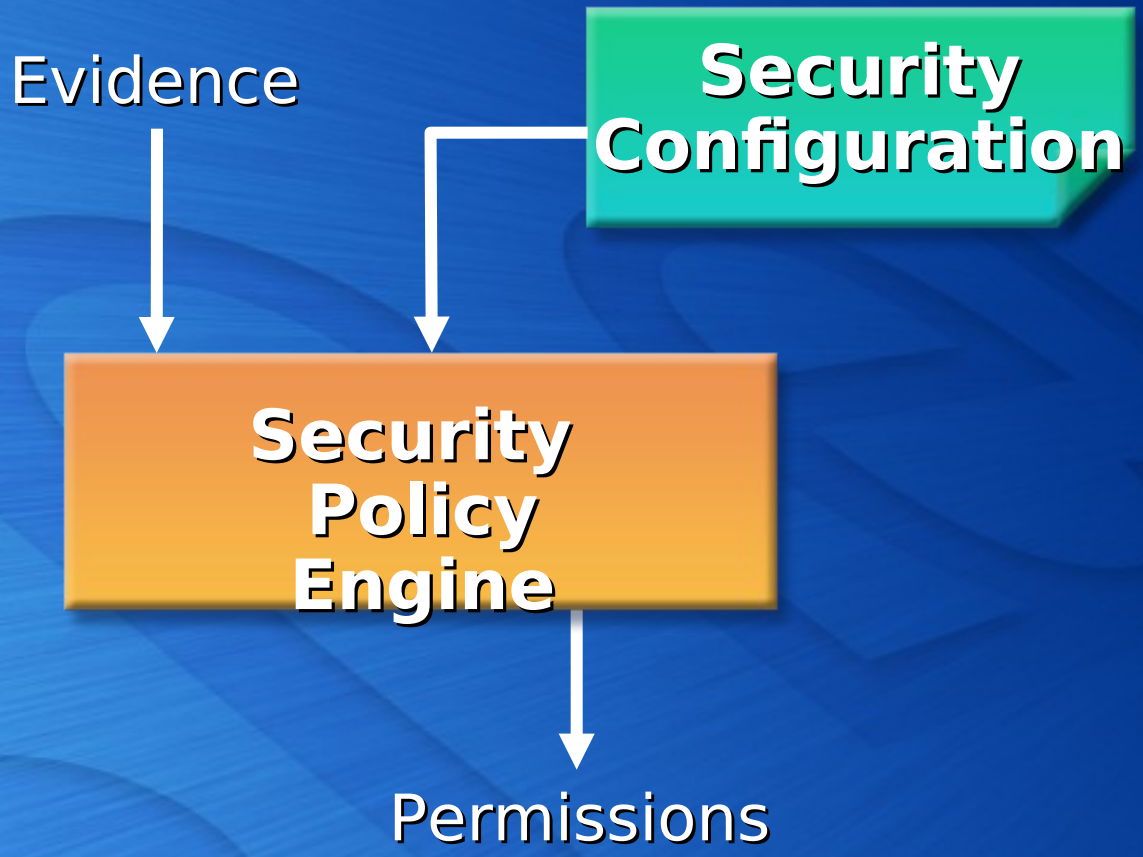
.NET Security

- Evidence-based Security
- Permissions
- Permission grants
- Security policy
- Principal
- Named permission sets
- Code Access Security
- Application domain hosts
- Code groups

Evidence-based Security

- Applies only to managed code
- Trust is determined by code evidence, not user identity
- Policy driven, generalized trust model using code evidence
- Fine-grained authorizations of code
- Additional layer of security over O/S
- Control over code sharing and authorizations

Evidence-based Security
Simplified Security Model



Evidence-based Security
Sources of Evidence

- Hosts specify "implicitly trusted" evidence
  - Shared names (crypto-sealed namespaces)
  - Publisher identity (AuthenticodeTM)
  - Code origin (URL, site and IE Zone)
- Assemblies specify "initially untrusted" evidence
  - Policy evaluates assembly evidence and decides whether to use it (e.g., third-party certifications)
- Assemblies can request permissions
  - Three types of requests: Minimal, optional and refuse
  - If policy does not grant everything in the

Permissions

- There are three kinds of permissions
  - Code Access permissions
  - Identity permissions
  - Role-based Security permissions

Permissions

- A permission object represents a specific authorization such as access to a resource
  - "permission to *do something*"
- A permission grant is an authorization given to an assembly (code)
  - "this code is authorized to *do something*."
- A permission demand is a security check for corresponding grants
  - "is *something* allowed?" (else, raise exception)

Permissions

- Code requests permissions
- Runtime grants permissions
- Code can demand that its callers have certain permissions
  - DNS permissions
  - Environment permissions
  - File I/O permissions
  - Registry permissions
  - More
- System.Security.Permissions Namespace

Permission Grants

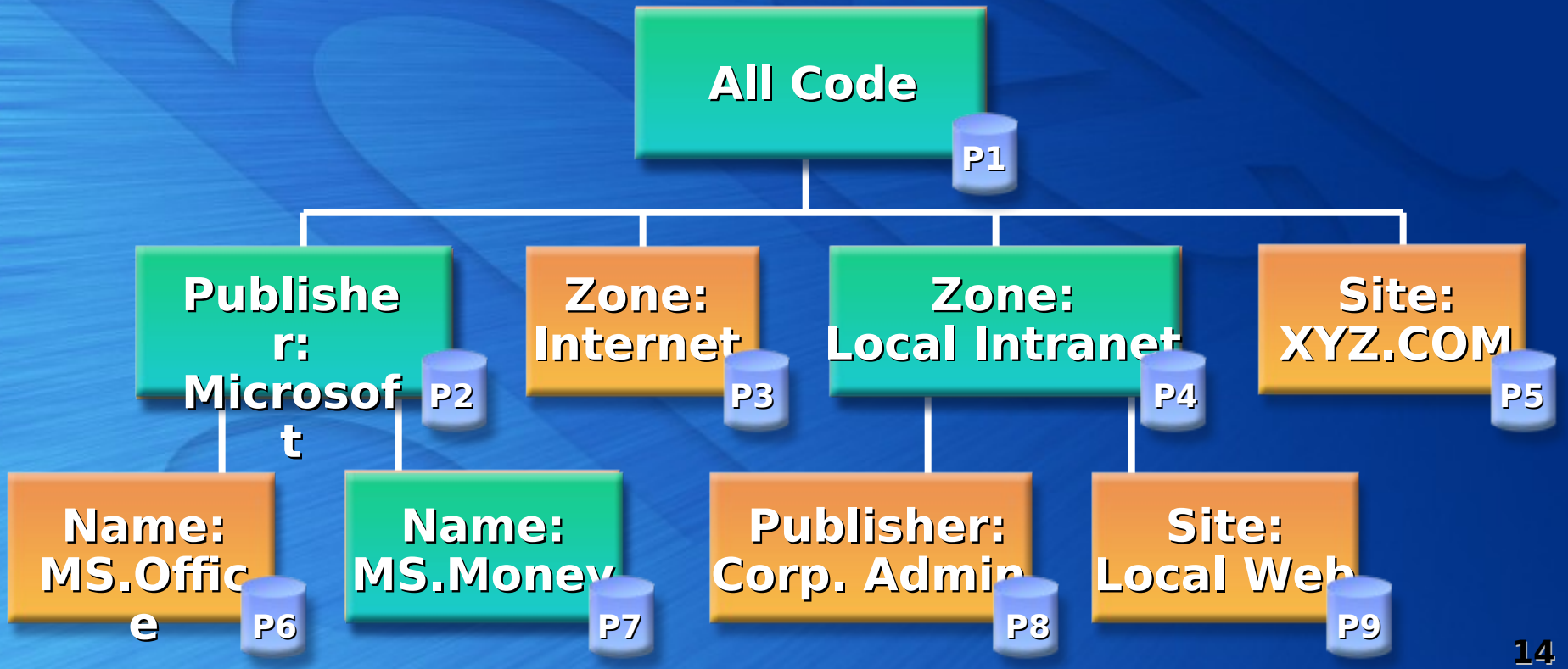
- Computing the allowed permission set
- Determining the granted permissions

Security Policy

- Configurable set of rules followed by the CLR in determining rights to grant code
- Administrators can use CASPOL to configure

Policy Level

- **Example: MS.Money on Local Intranet**
  - **Member of four groups (highlighted)**
  - **Granted permissions = P1 ∪ P2 ∪ P7 ∪ P4**



Code Group

- Logical grouping of code
- Code can belong to one or more groups
- All code belongs to “all code” group
- Membership in group determined by **Evidence**

Principal

- A user or something acting on the user's behalf
- Generic Principals - Unauthenticated
- Windows Principals - Windows users
- Custom Principals - Defined by application

Named Permission Sets

- Associated with code groups
- Custom sets can be defined by administrators - simplified administration
- Nothing, Execution, Internet, LocalIntranet, Everything and FullTrust

Code Access Security

- Security enforceable on well-behaved code
- Code is verified to be memory type safe
  - Can only get to objects to which it has references
  - Can only use defined interfaces to objects
- Code with skip-verify permission is excepted
- Verification rules are safe, but not complete
- Verifiability is compiler code issue

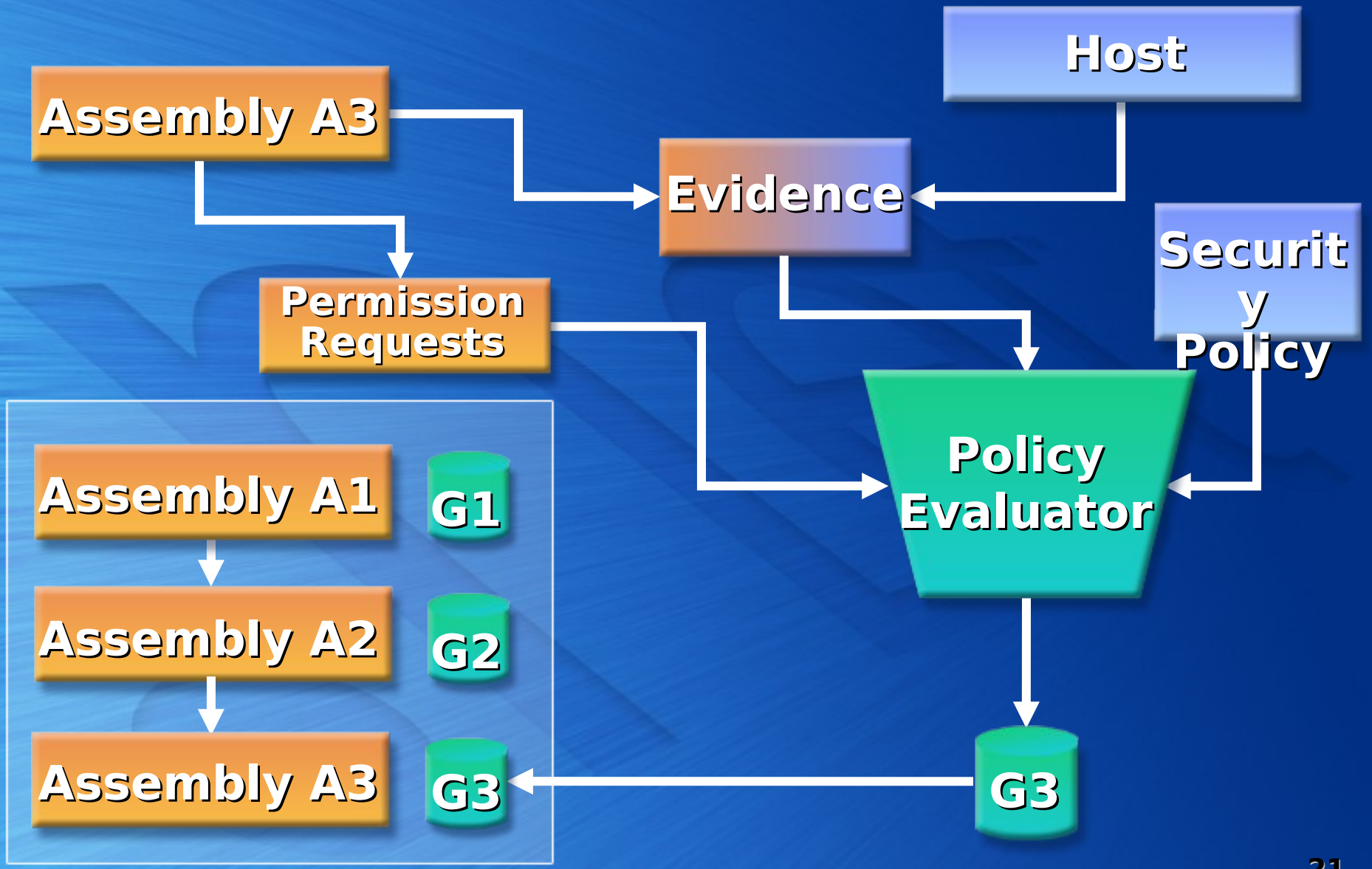
Code Access Security

- Feature of CLR
- Verify code's identity and its origins
- Specify the operations that the code is allowed to perform
- Enable mobile code usage

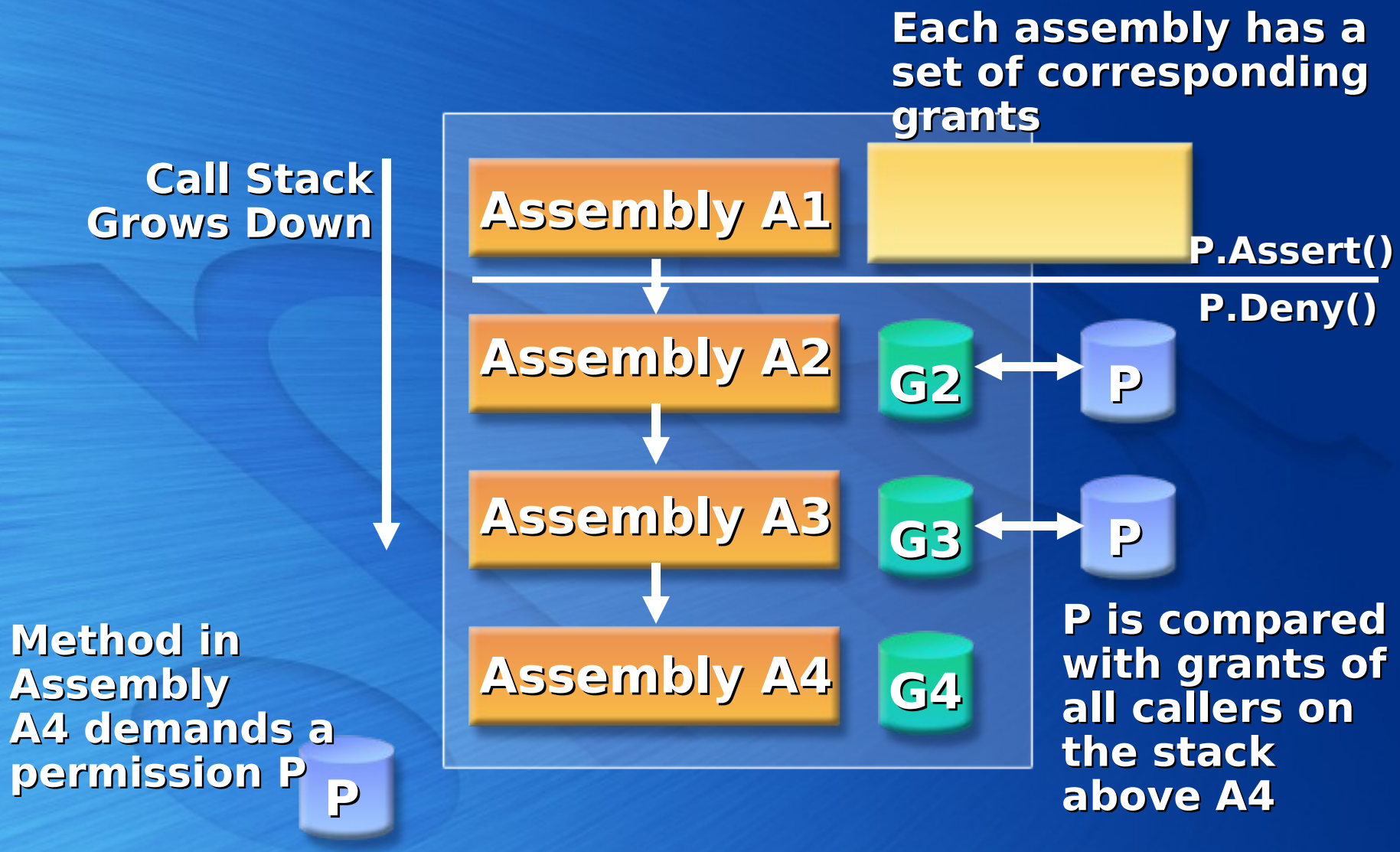
Code Access Security

- Based on causality of code execution
- Most permissions are Code Access permissions
  - Demanding a permission performs a stack walk checking for related grants of all callers
  - Support dynamic stack modifiers
  - Two ways to make checks:
    - Imperatively (method implementation)
    - Declaratively (method metadata)

Putting It Together



Checking Permissions



Application Domain Hosts

- Each application operates in an “application domain”
- Types of hosts:
  - Shell host
  - Browser host
  - Server host
  - Custom host

Miscellaneous Topics

- Cryptography Namespace
- Code Obfuscation
- Type Safety
- Security Tools

Cryptography Namespace

- System.Security.Cryptography
- Provides cryptographic services
  - Random Numbers, Symmetric (DES, 3DES, RC2) and Asymmetric (RSA) Algorithms, Hashing (SHA1, MD5)
- Wraps Crypto API
  - Properly developed
  - Highly functional
  - Extensible

Code Obfuscation

- As with any program representation, MSIL can be decompiled
- Raises intellectual property issues
- Obfuscation makes decompiled code harder to analyze for both humans and software

Type Safety

- Isolates assemblies
- Prevents direct calls into native code
- Policy can allow code to bypass

Security Tools

- The graphical tool for managing security policy is MSCORCFG. It is an MMC snap-in
- CASPOL, cert2spc, Certmgr, Chktrust, permview, peverify, secutil, signcode

Summary

- **Evidence-based Security**
  - Policy model for granting permissions
  - Lots of new functionality and flexibility for clients and servers
  - Extensibility and customization available throughout the model
- **Code Access Security**
  - Authorization enforcement mechanism for partially trusted code

Resources

- ***How ASP Security Works***
  An overview of ASP Security
  http://msdn.microsoft.com/library/dotnet/cpguide/cpconhowaspnetsecurityworks.htm
- ***Key Concepts in Application Security***
  A basic review of the major components needed to secure applications
  http://msdn.microsoft.com/library/dotnet/cpguide/cpconkeyconceptsinsecurity.htm

Microsoft®